Modelo de IA para el diagnóstico temprano de enfermedades en maíz a partir de imágenes de hojas

GRUPO N. 7

Emersson Elian Lopez Pita - 2225507
Cesar Alejandro Soto Paipa - 2225612
Juan David Mena Gamboa - 2221886

www.reallygreatsite.com

LET'S START

GRUPO N. 7

Problematica

- Cultivos clave para economía y seguridad alimentaria

- Enfermedades afectan la productividad

- Diagnóstico actual depende de expertos → costoso y poco accesible

NEXT SLIDE

Objetivo del modelo IA

- Detectar si una hoja está enferma
- Identificar la enfermedad específica
- Trabajo enfocado en una especie/enfermedad concreta

NEXT SLIDE

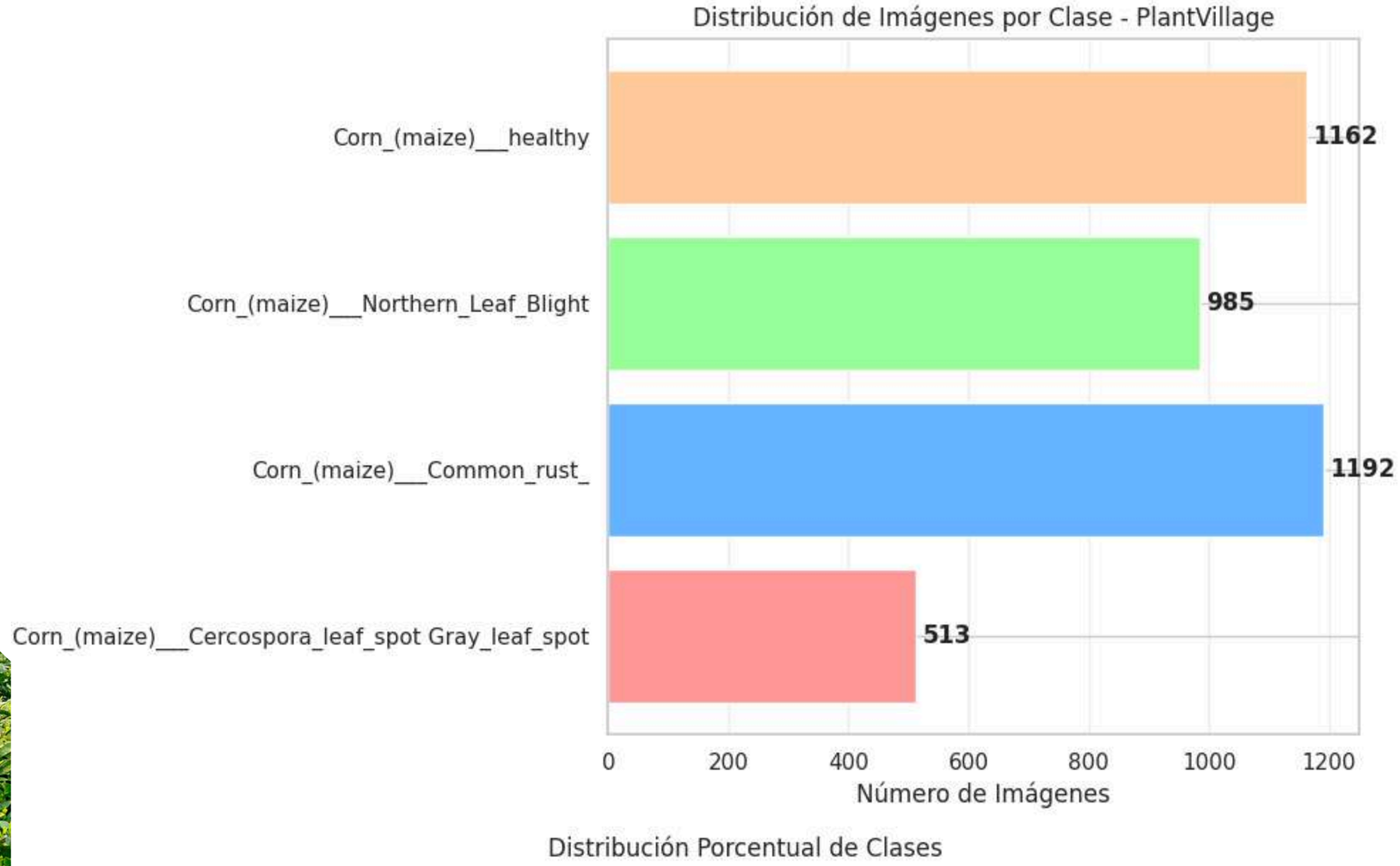
Cómo funcionará el modelo

- **Paso 1: La IA compara la imagen con ejemplos que ya tiene guardados el dataset**
- **Paso 2: El sistema dice si la hoja está sana o enferma.**
- **Paso 3: Si está enferma, indica qué enfermedad específica es.**

NEXT SLIDE

Dataset

GRUPO N. 7



100%

NEXT SLIDE

Dataset

Distribución Porcentual de Clases

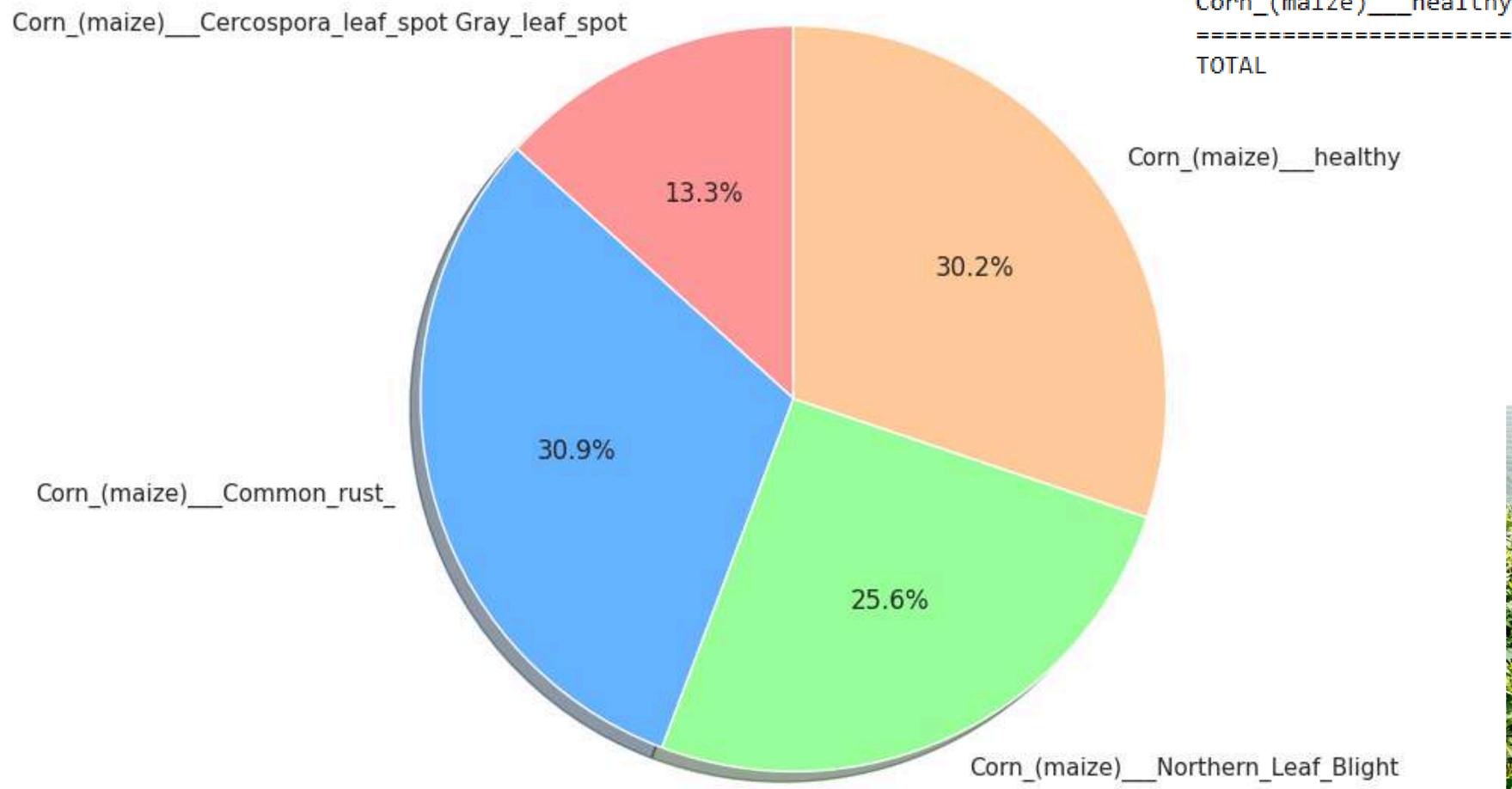


TABLA RESUMEN DEL DATASET:

CLASE	IMÁGENES	%
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot	513	13.32
Corn_(maize)___Common_rust_	1192	30.94
Corn_(maize)___Northern_Leaf_Blight	985	25.57
Corn_(maize)___healthy	1162	30.17
TOTAL	3852	100%

100%

NEXT SLIDE

Clases

Hoja Sana

Common Rust

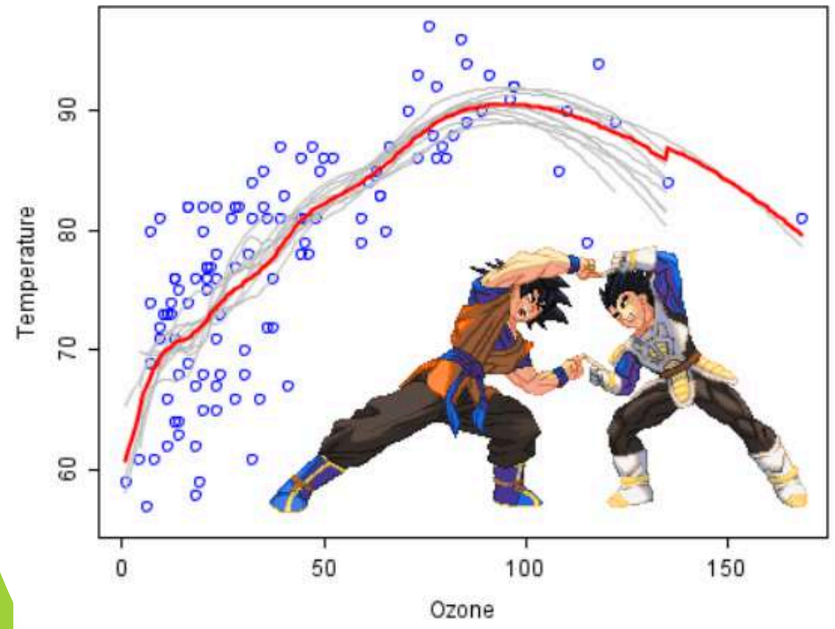
Northern Leaf Blight

Cercospora Leaf Spot (Gray Leaf Spot)

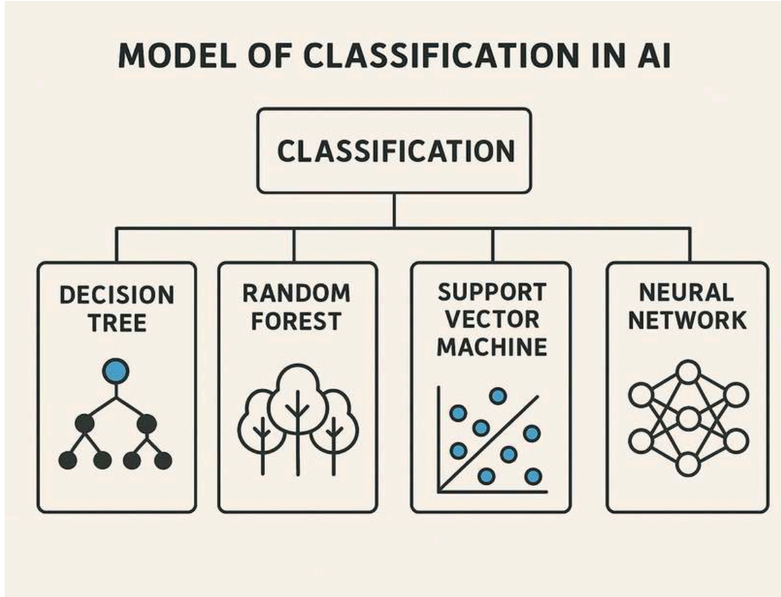
Problema de modelo

GRUPO N. 7

- Modelo de Regresion



- Modelo de Clasificación



NEXT SLIDE

Analisis con ML

- **Preparación de datos y extracción de caracteristicas**

```
img_path = os.path.join(folder, file)
img = Image.open(img_path).convert("RGB").resize((64,64))
arr = np.array(img)/255.0
mean = arr.mean(axis=(0,1))
std = arr.std(axis=(0,1))
features = np.concatenate([mean,std])
data.append([*features, label])
```

- **Distribución de las clases**

```
Cantidad de imágenes por clase:
label
Corn_(maize)___Common_rust_                    1192
Corn_(maize)___healthy                         1162
Corn_(maize)___Northern_Leaf_Blight             985
Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot  513
```

NEXT SLIDE

Train Test

```
X = df.drop('label', axis=1)
y = df['label']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)

scaler = StandardScaler() # Esto es para normalizar
X_train_scaled = scaler.fit_transform(X_train)
X_test_scaled = scaler.transform(X_test)

print("Tamaño train:", X_train.shape, " | Tamaño test:", X_test.shape)
```

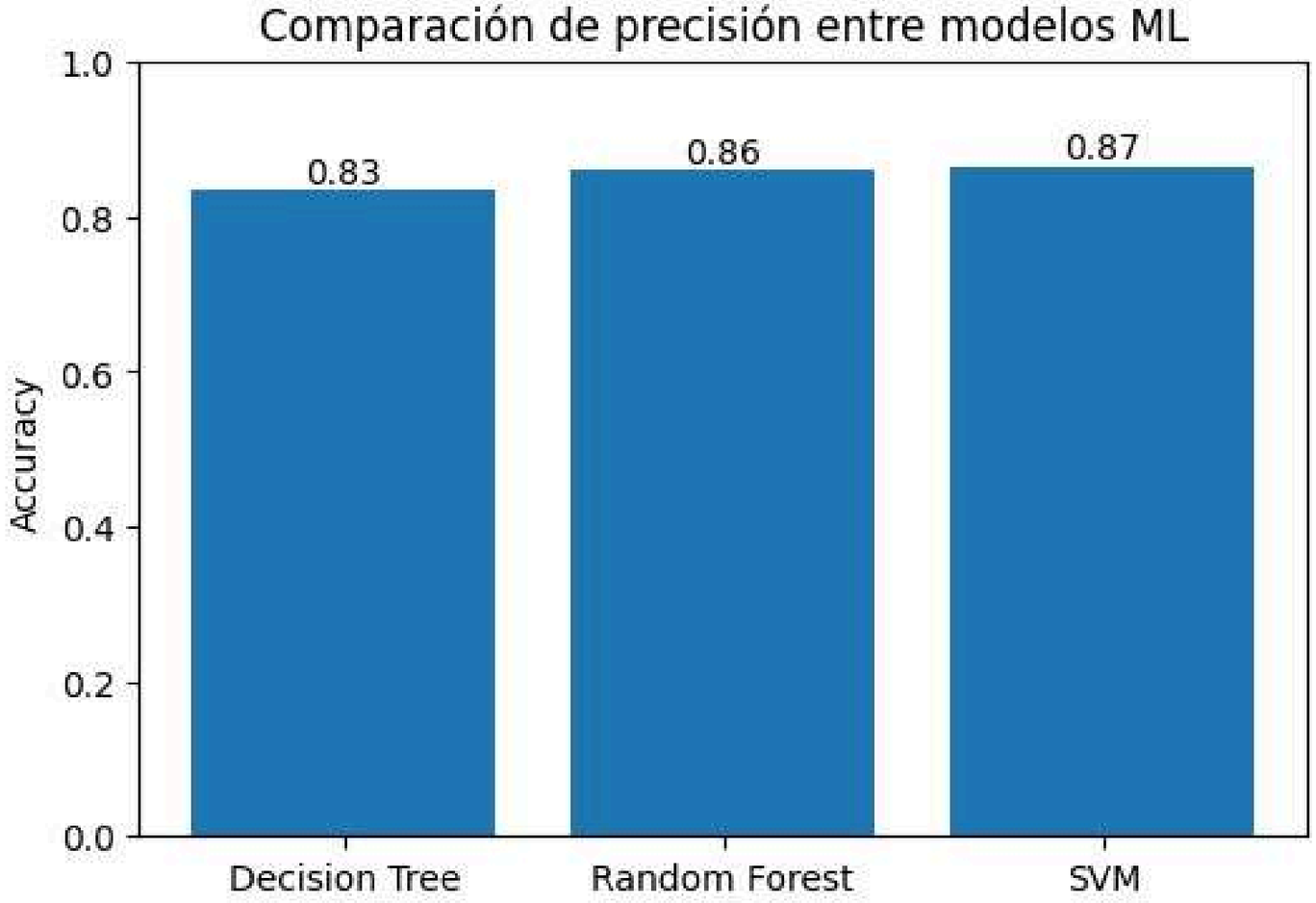
```
Tamaño train: (2696, 6)  | Tamaño test: (1156, 6)
```

- **Modelos comparados**

```
models = {
    "Decision Tree": DecisionTreeClassifier(),
    "Random Forest": RandomForestClassifier(),
    "SVM": SVC()
}
```

NEXT SLIDE

Comparación de Accuracy



NEXT SLIDE

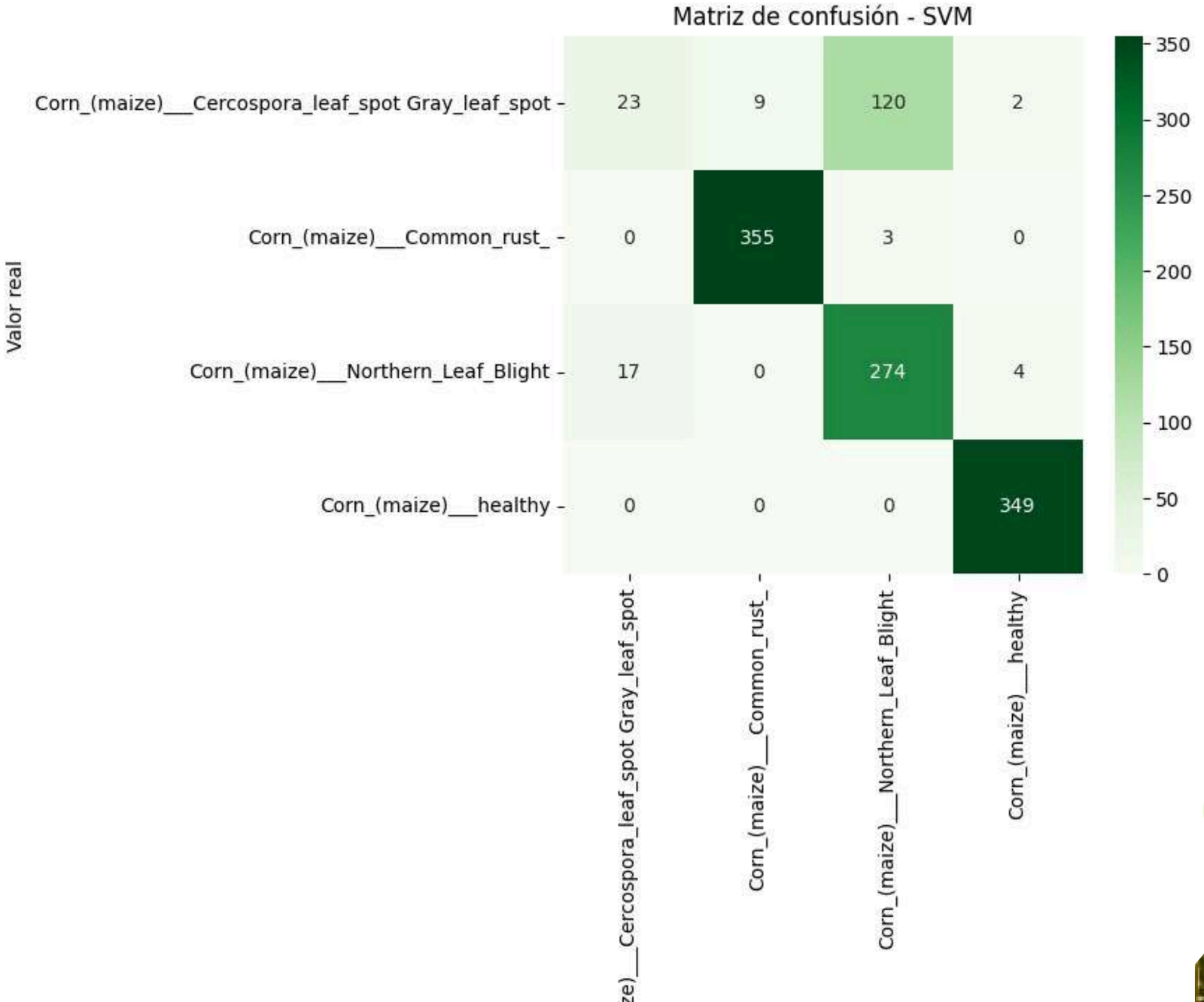
K-Fold

```python
kf = KFold(n_splits=5, shuffle=True, random_state=42)

for name, model in models.items():
    scores = cross_val_score(model, X_train_scaled, y_train, cv=kf, scoring='accuracy')
    print(f"{name} | Accuracy promedio (5-fold): {scores.mean():.3f} ± {scores.std():.3f}")
```

```
Decision Tree | Accuracy promedio (5-fold): 0.812 ± 0.014
Random Forest | Accuracy promedio (5-fold): 0.861 ± 0.006
SVM | Accuracy promedio (5-fold): 0.872 ± 0.010
```

NEXT SLIDE

Matriz de confusion

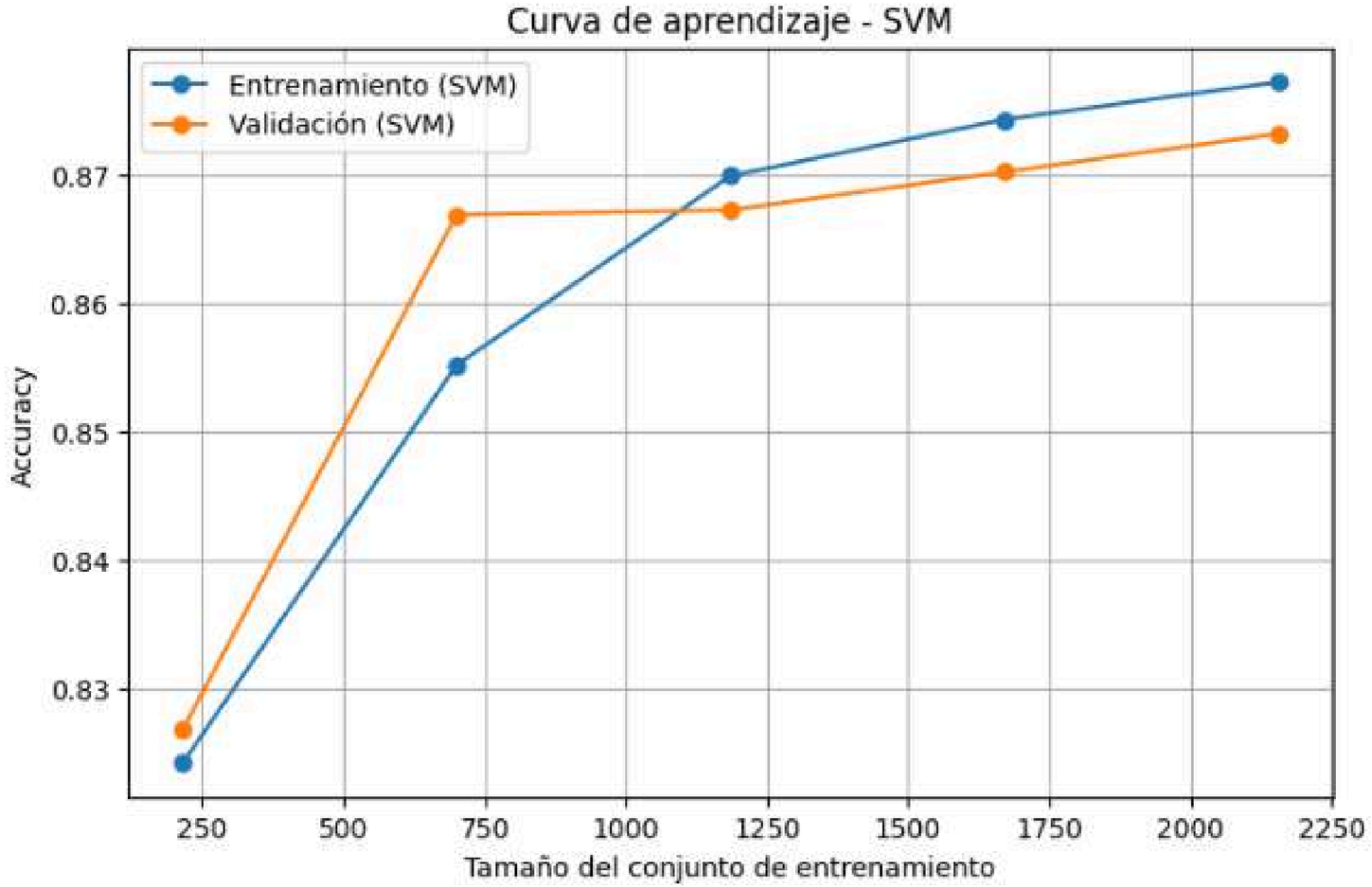


GRUPO N. 7

NEXT SLIDE

Curva de aprendizaje

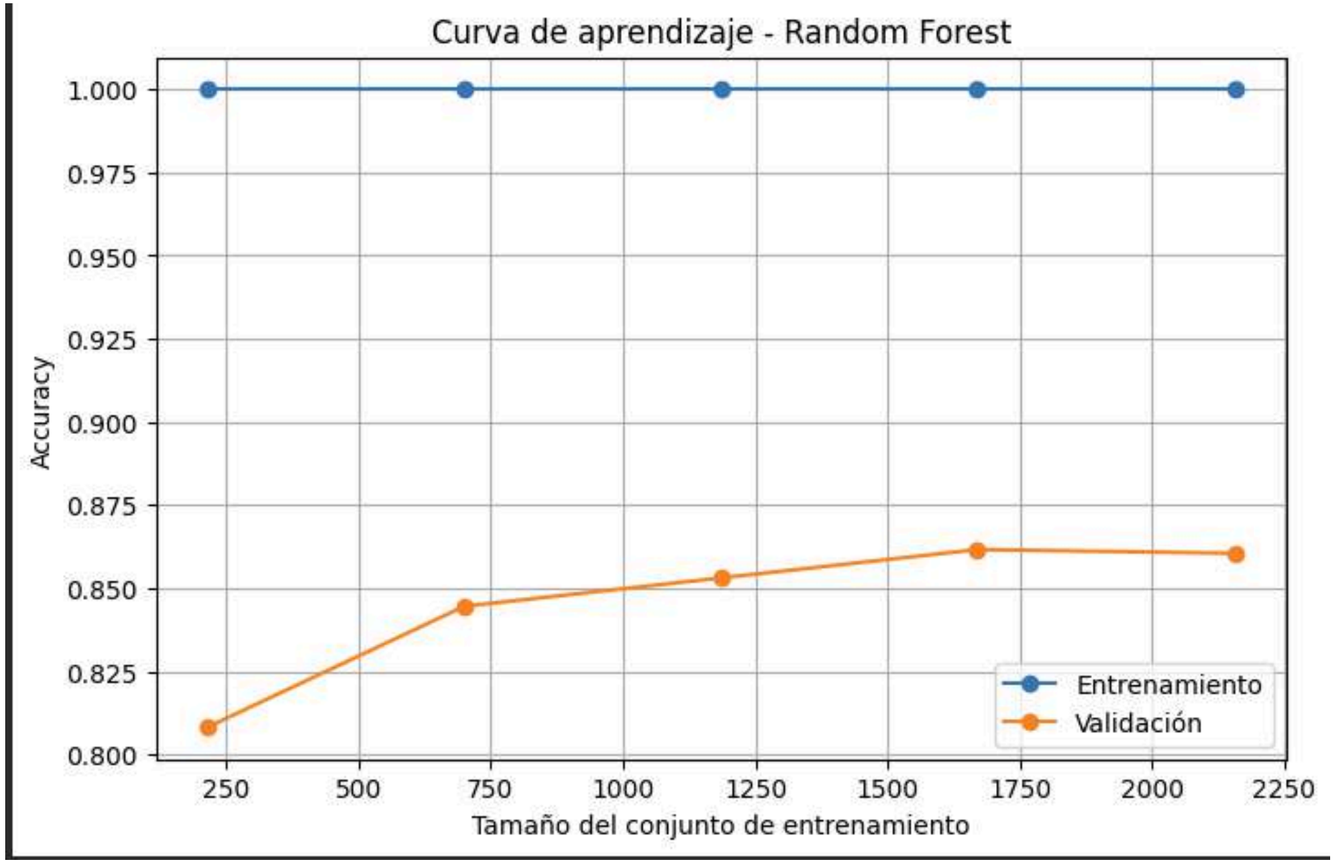
GRUPO N. 7



NEXT SLIDE

Curva de aprendizaje

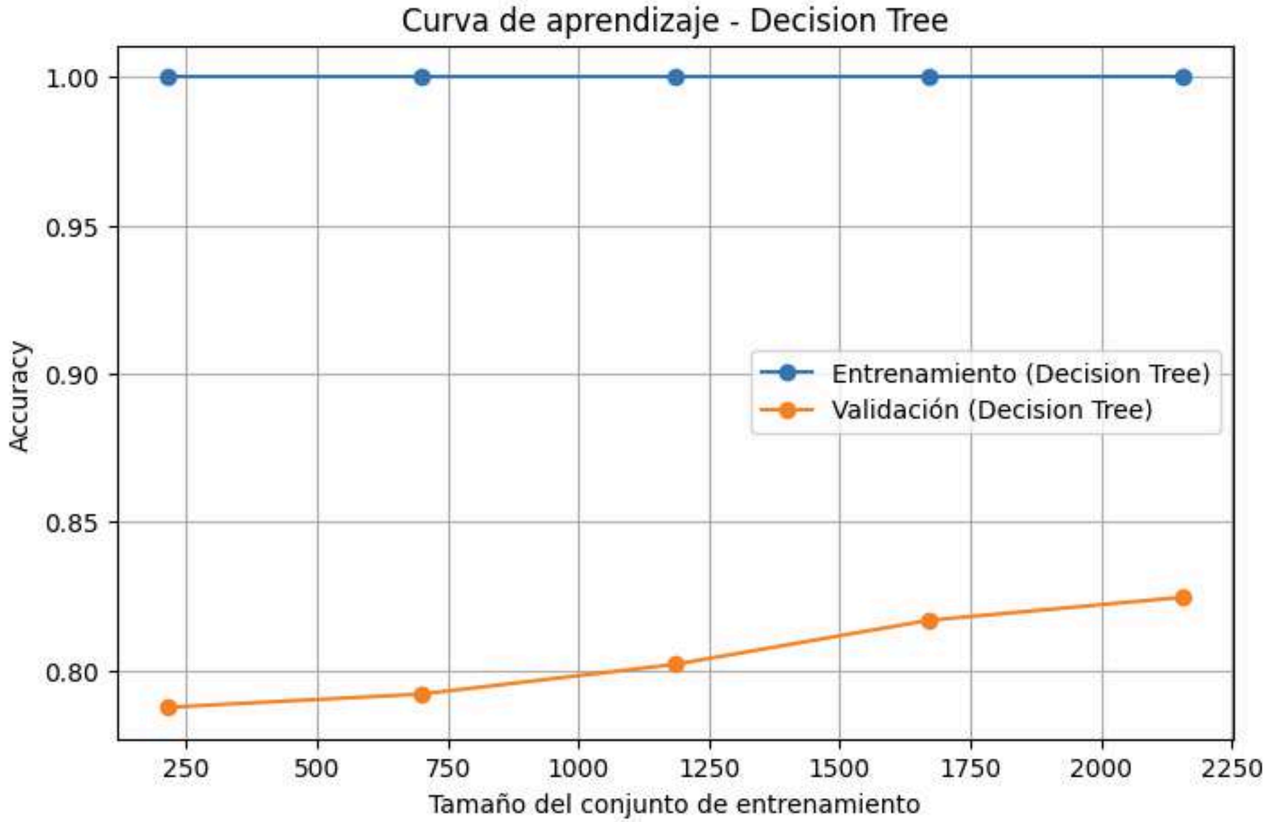
GRUPO N. 7



NEXT SLIDE

Curva de aprendizaje

GRUPO N. 7



NEXT SLIDE

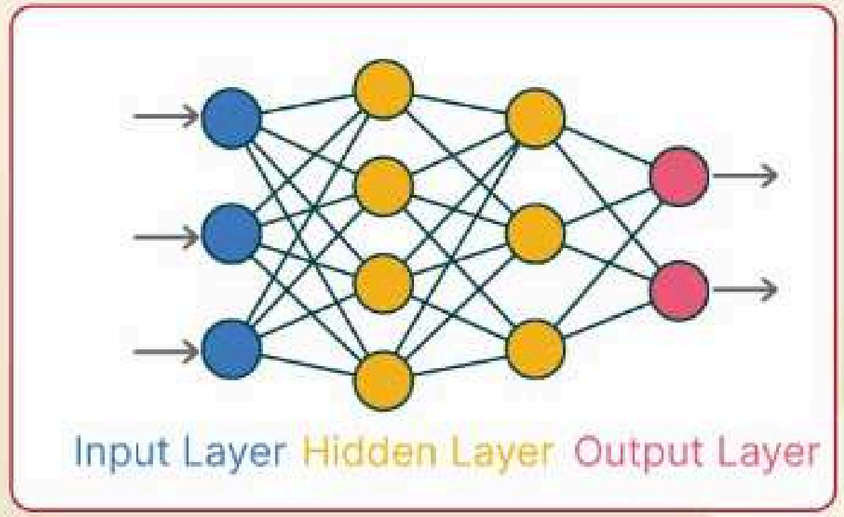
MOVING FROM ML → DEEP LEARNING

GRUPO N. 7

LET'S START

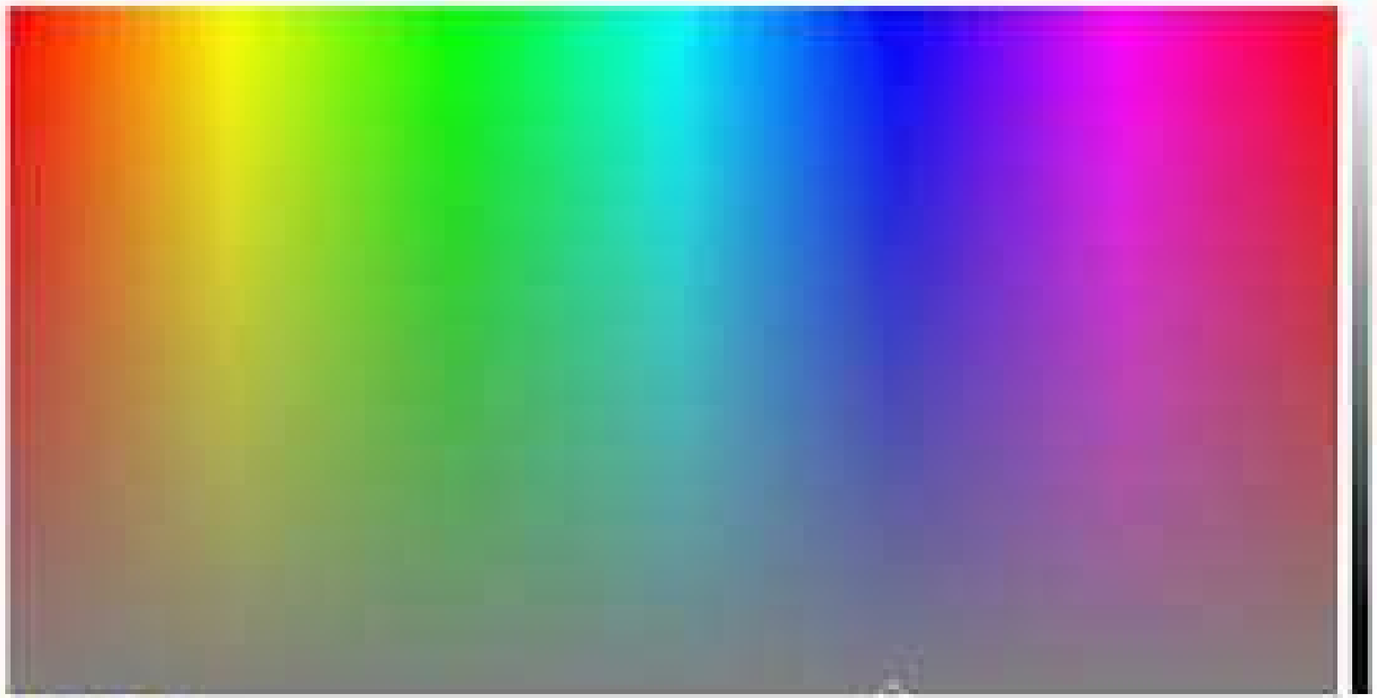
Enfoque con Deep Learning

Multilayer Perceptron (MLP) Neural Networks



Estructura de una imagen digital

INGOUDE COMPANY



IONOS

Espacio de color RGB

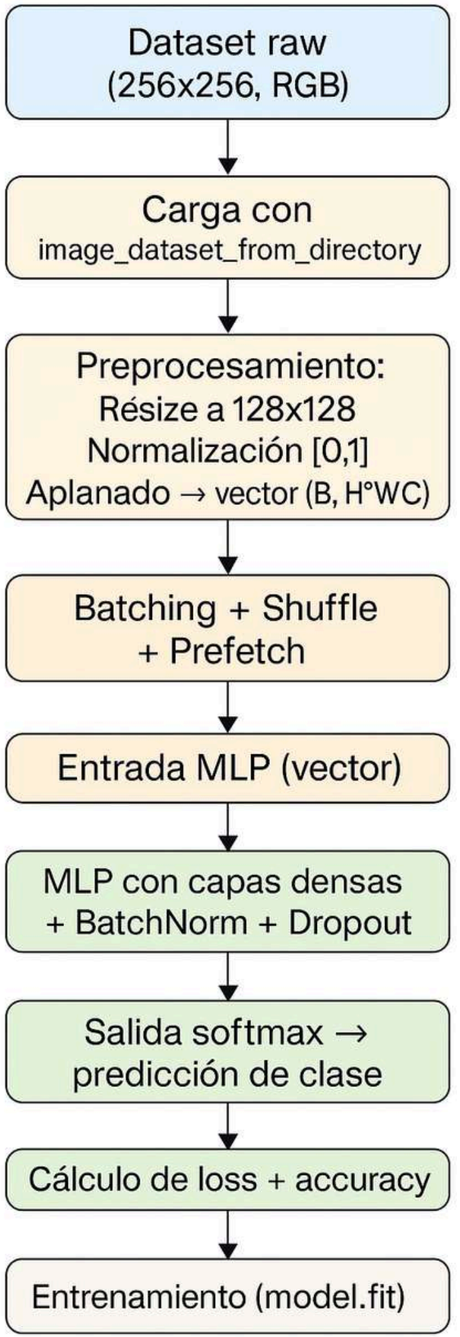
“Estas matrices son la entrada del modelo de Deep Learning. Mientras los modelos de Machine Learning tradicional necesitan extraer características manualmente (color, textura, forma), el modelo de red neuronal aprende directamente de los valores de estos píxeles.”

Verificación de Dimensiones de Imágenes

```python
import os
from PIL import Image

# Obtener lista de archivos de imagen
extensiones_validas = ('.jpg', '.jpeg', '.png', '.bmp', '.tiff', '.JPG', '.JPEG', '.PNG')
archivos_imagen = [f for f in os.listdir(directorio) if f.endswith(extensiones_validas)]

if not archivos_imagen:
    return {"error": "No se encontraron archivos de imagen en el directorio"}

# Verificar cada imagen
dimensiones_encontradas = set()
imagenes_problematicas = []
total_imagenes = len(archivos_imagen)
```

Entrenamiento del Modelo Clasificador

```
Found 3852 files belonging to 4 classes.
Using 3082 files for training.
Found 3852 files belonging to 4 classes.
Using 770 files for validation.
Orden de clases (fijo): ['Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot', 'Corn_(maize)___Common_rust_', 'Corn_(maize)__
_healthy', 'Corn_(maize)___Northern_Leaf_Blight']
Distribución clases (train): {'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot': 427, 'Corn_(maize)___Common_rust_': 942,
'Corn_(maize)___healthy': 950, 'Corn_(maize)___Northern_Leaf_Blight': 763}
Class weights: {0: 7.217798594847775, 1: 3.2717622080679405, 2: 3.2442105263157894, 3: 4.039318479685452}
Epoch 1/10
97/97 ━━━━━━━━━━━━━━━━━━━━ 51s 497ms/step - accuracy: 0.7579 - loss: 3.0762 - val_accuracy: 0.1558 - val_loss: 2.7370 - learn
ing_rate: 0.0010
Epoch 2/10
97/97 ━━━━━━━━━━━━━━━━━━━━ 47s 489ms/step - accuracy: 0.8381 - loss: 2.2144 - val_accuracy: 0.6727 - val_loss: 1.1200 - learn
ing_rate: 0.0010
Epoch 3/10
97/97 ━━━━━━━━━━━━━━━━━━━━ 83s 494ms/step - accuracy: 0.8621 - loss: 2.0062 - val_accuracy: 0.7974 - val_loss: 0.9811 - learn
ing_rate: 0.0010
```



Evaluación y Predicción de Imágenes con el Modelo Entrenado

GRUPO N. 7


```python
reg = tf.keras.regularizers.l2(1e-4)
model = tf.keras.Sequential([
    # Primera capa necesita input_shape
    tf.keras.layers.BatchNormalization(input_shape=(target_size[0]*target_size[1]*3,)),
    tf.keras.layers.Dense(1024, activation="relu", kernel_regularizer=reg),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.4),
    tf.keras.layers.Dense(512, activation="relu", kernel_regularizer=reg),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.3),
    tf.keras.layers.Dense(128, activation="relu", kernel_regularizer=reg),
    tf.keras.layers.BatchNormalization(),
    tf.keras.layers.Dropout(0.2),
    tf.keras.layers.Dense(num_classes, activation="softmax")
])
```

NEXT SLIDE

Algunas Pruebas

INGOUDE COMPANY

```
test_loss, test_acc = model_infer.evaluate(val_ds_raw, verbose=1)
```

```
25/25 ━━━━━━━━━━━━━━━━━━━━ 2s 70ms/step - accuracy: 0.9273 - loss: 0.7095
Test loss: 0.7095332741737366
Test accuracy: 0.9272727370262146
```

Algunas Pruebas

INGOUDE COMPANY

In [12]:

```
dataset_path = "plantvillage dataset/color"
vegetable_leaf = "Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot"
specific_img = "00a20f6f-e8bd-4453-9e25-36ea70feb626___RS_GLSp 4655.JPG"
early_blight = os.path.join(dataset_path, vegetable_leaf, specific_img)
# Ejecuto
predict_external_image(early_blight)
```

Pred: Corn_(maize)___Northern_Leaf_Blight (p=0.88)

```
Probs: {'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot': 0.1215328574180603, 'Corn_(maize)___Common_rust_': 0.0001279489
0790246427, 'Corn_(maize)___healthy': 1.767093999660574e-05, 'Corn_(maize)___Northern_Leaf_Blight': 0.8783215284347534}
```

Out[12]:

```
('Corn_(maize)___Northern_Leaf_Blight',
 array([1.2153286e-01, 1.2794891e-04, 1.7670940e-05, 8.7832153e-01],
       dtype=float32))
```

Algunas Pruebas

INGOUDE COMPANY

In [13]:

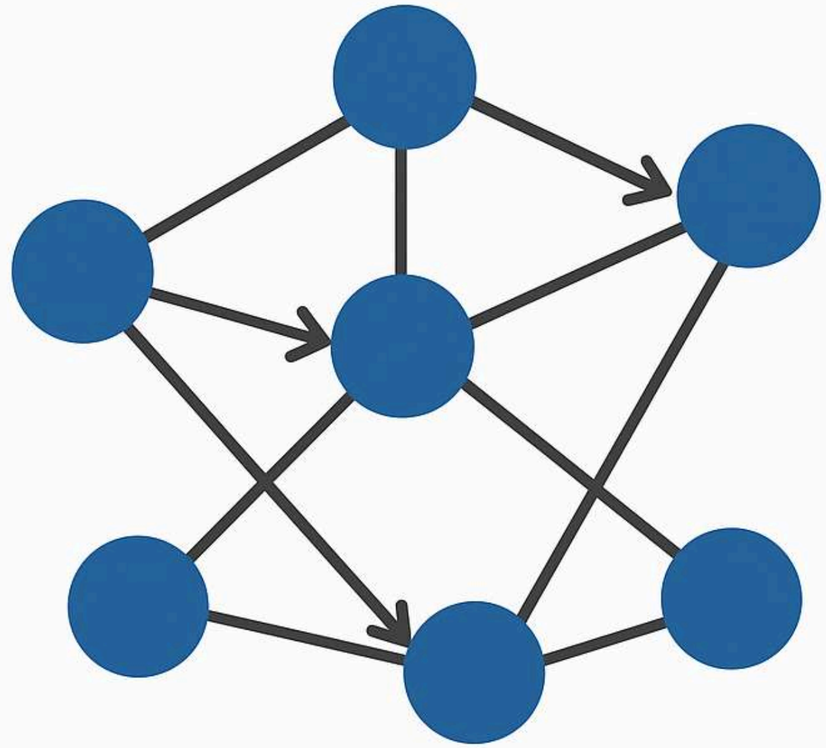
```
dataset_path = "plantvillage dataset/color"
vegetable_leaf = "Corn_(maize)___Common_rust_"
specific_img = "RS_Rust 1563.JPG"

late_blight = os.path.join(dataset_path, vegetable_leaf, specific_img)

predict_external_image(late_blight)
```


```
Probs: {'Corn_(maize)___Cercospora_leaf_spot Gray_leaf_spot': 2.6803040782397147e-06, 'Corn_(maize)___Common_rust_': 0.999997
1389770508, 'Corn_(maize)___healthy': 1.9731347222773366e-08, 'Corn_(maize)___Northern_Leaf_Blight': 9.820931978765657e-08}
```

Out[13]:

```
('Corn_(maize)___Common_rust_',
 array([2.6803041e-06, 9.9999714e-01, 1.9731347e-08, 9.8209320e-08],
       dtype=float32))
```

LENGUAJE NO SUPERVISADO



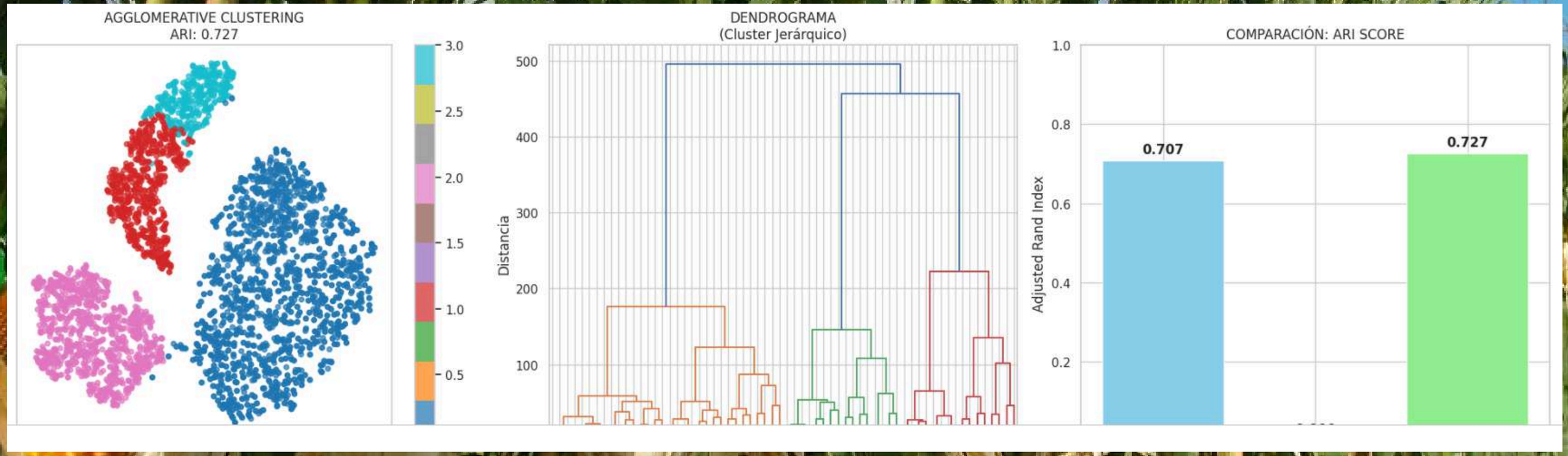
Metodos Usados

GRUPO N. 7

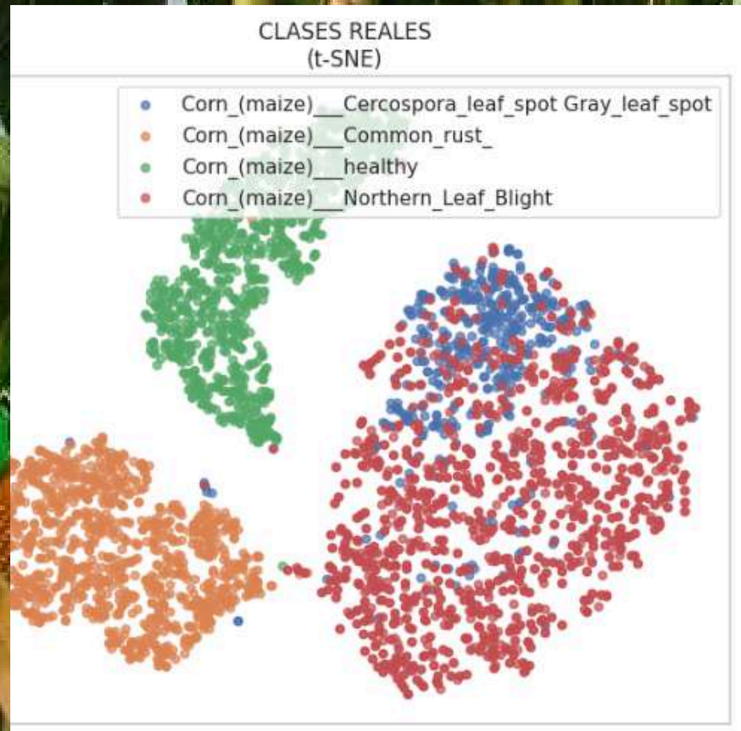
```
=== EVALUACIÓN CON MÉTRICAS ===
 EVALUANDO MÉTRICAS DE CALIDAD...
 KMEANS:
   - Clusters encontrados: 4
   - Adjusted Rand Index: 0.707
   - Silhouette Score: 0.471
 DBSCAN:
   - Clusters encontrados: 1
   - Adjusted Rand Index: 0.000
   - Silhouette Score: -1.000
 AGGLOMERATIVE:
   - Clusters encontrados: 4
   - Adjusted Rand Index: 0.727
   - Silhouette Score: 0.461

 MEJOR MÉTODO: Agglomerative (ARI: 0.727)
```

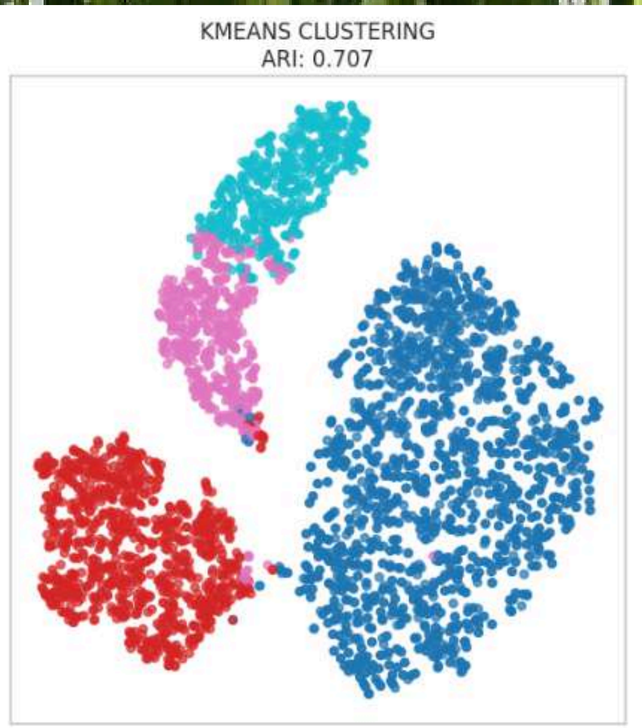
NEXT SLIDE

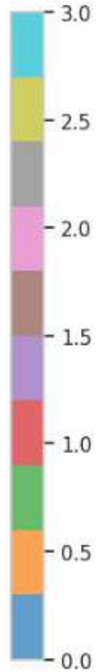
Representacion Grafica

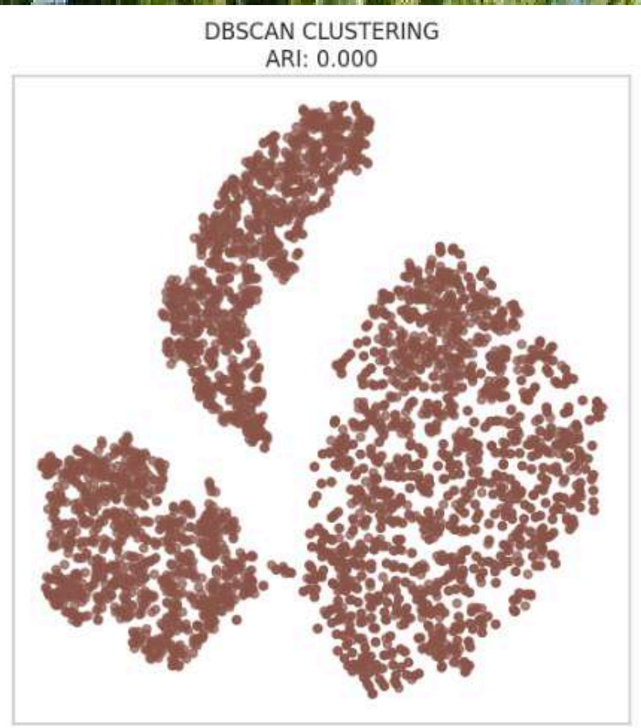


Representacion Grafica

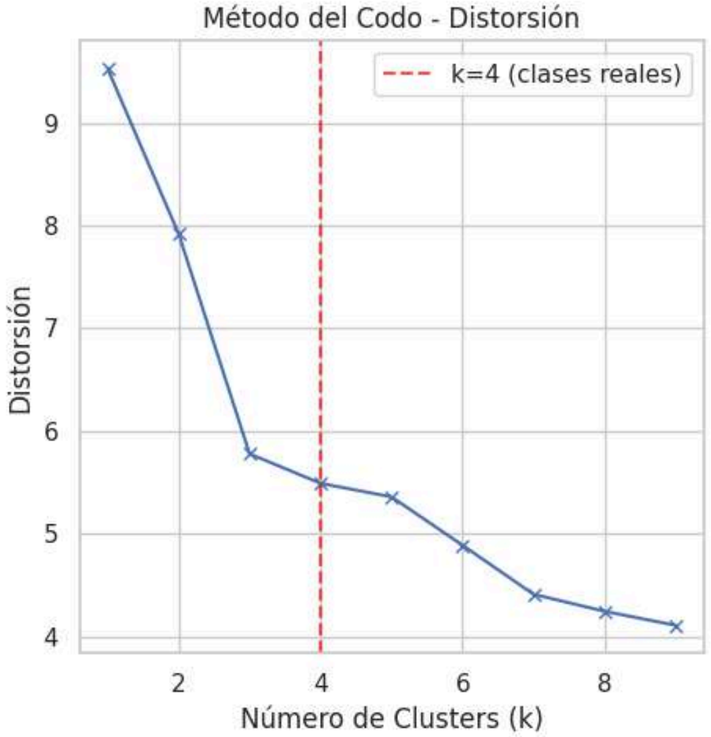


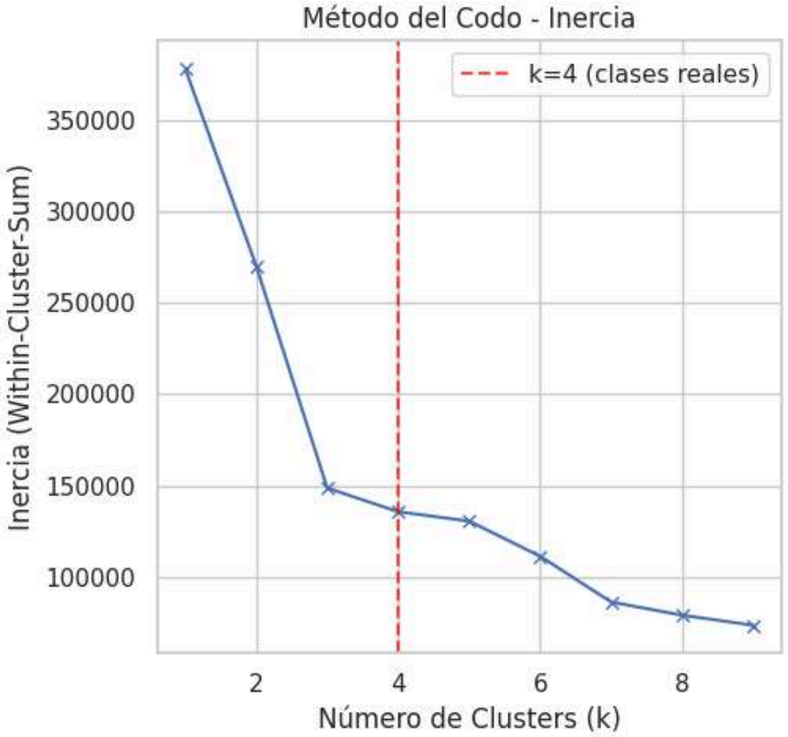


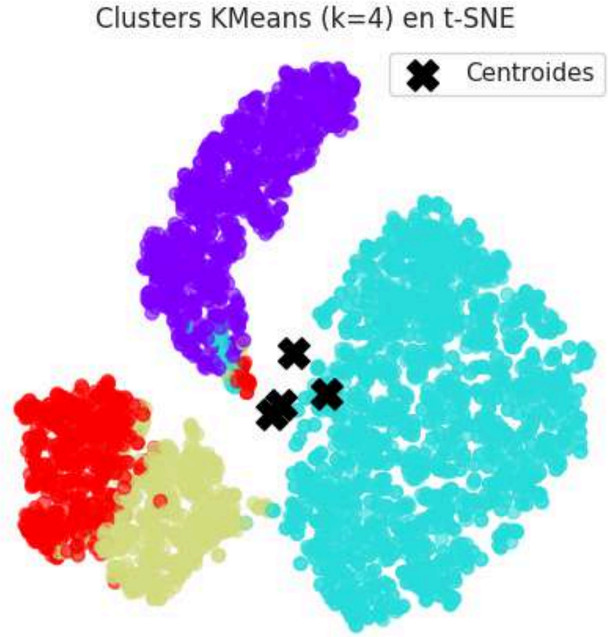




Metodo del codo







INTERPRETACIÓN DEL MÉTODO DEL CODO:

- Inercia con k=4: 135712.28
- ¿El codo sugiere k=4? → NO - el codo sugiere otro k
  Pero k=4 está justificado por las clases biológicas reales

Thank you for your attention